Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A system for verification of a system design, comprising:

a test program generator that accepts a sequence of statements including at least one event;

an event handling facility in said test program generator, wherein said event includes a plurality of events that are processed in order of priority values thereof in said event handling facility; and

wherein responsive to a triggering condition of said event said test program generator emits test program instructions in response to one of a primary input stream and an alternate input stream, said alternate input stream being represented in a body of said event.

2. (Canceled)

3. (Currently amended) The system according to claim 1, wherein a conditional statement of each of said ~~event~~ events references a current state of a test program that is generated by said test program generator.

4. (Currently amended) A method of test program generation, comprising the steps of:

defining a set of statements, said set of statements including an event;

responsive to said set of statements generating a sequence of test program instructions for a target;

while performing said step of generating said sequence of test program instructions determining if a condition of said event is satisfied, wherein said event comprises a plurality of events, each of said events having a priority value, and said step of determining if said condition is satisfied is performed with respect to each of said events in an order of said priority value thereof; and

responsive to said step of determining generating an alternate sequence of test program instructions.

5. (Original) The method according to claim 4, wherein said step of determining is performed by evaluating a state of said target prior to inclusion of an instruction in said sequence of test program instructions.

6. (Original) The method according to claim 4, wherein at least a portion of said sequence of test program instructions are randomly generated.

7. (Canceled)

8. (Currently amended) The method according to claim 4, wherein each of said ~~event~~ events ~~has~~ have an identifying name attribute.

9. (Currently amended) The method according to claim 4, wherein each of said ~~event~~ events has a triggering condition attribute.

10. (Currently amended) The method according to claim 4, wherein each of said ~~event~~ events comprises an input stream entity.

11. (Currently amended) ~~The method according to claim 4,~~ A method of test program generation, comprising the steps of:

defining a set of statements, said set of statements including an event; wherein said event comprises an identifying name attribute, a triggering condition attribute, a priority value and an input stream entity;

responsive to said set of statements generating a sequence of test program instructions for a target;

while performing said step of generating said sequence of test program instructions determining if a condition of said event is satisfied; and

responsive to said step of determining generating an alternate sequence of test program instructions.

12. (Original) A method of test program generation, comprising the steps of:

providing a test program generation engine;

coupling said test program generation engine to a design specification of a target, wherein said design specification comprises a knowledge base;

coupling said test program generation engine to an architectural simulator of said target;

introducing a set of statements into said test program generation engine, said set of statements including an event;

determining whether a triggering condition of said event is satisfied;

in a first case, wherein said triggering condition is not satisfied, causing said test program generation engine to respond to said set of statements to generate a first sequence of test program instructions that can be executed on said target; and

in a second case, wherein said triggering condition is satisfied, causing said test program generation engine to respond to an alternate set of statements of said event to generate a second sequence of test program instructions that can be executed on said target.

13. (Original) The method according to claim 12, wherein at least a portion of said first sequence of test program instructions and said second sequence of test program instructions is generated randomly.

14. (Original) The method according to claim 12, wherein said step of determining is performed by evaluating a state of said target prior to inclusion of an instruction in said first sequence of test program instructions.

15. (Original) The method according to claim 12, wherein said event comprises a plurality of events, each of said events having a priority value, and said step of determining is performed with respect to each of said events in an order of said priority value thereof.

16. (Original) The method according to claim 12, wherein said set of statements is introduced into said test program generation engine as an input file.

17. (Original) The method according to claim 12, wherein said event has an identifying name attribute.

18. (Original) The method according to claim 12, wherein said event has a triggering condition attribute.

19. (Original) The method according to claim 12, wherein said event comprises an input stream.

20. (Original) The method according to claim 12, wherein said event comprises an identifying name attribute, a triggering condition attribute, a priority value and an input stream.

21. (Currently amended) A computer software product, comprising a computer-readable medium in which computer program instructions are stored, which instructions, when read by a computer, cause the computer to generate test programs by performing the steps of:

accepting a set of statements, said set of statements including an event;

responsive to said set of statements generating a sequence of test program instructions for a target;

while performing said step of generating said sequence of test program instructions determining if a condition of said event is satisfied wherein said event comprises a plurality of events, each of said events having a priority value, and said step of determining if said condition is satisfied is performed with respect to each of said events in an order of said priority value thereof; and

responsive to said step of determining generating an alternate sequence of test program instructions.

22. (Original) The computer software product according to claim 21, further comprising the steps of accessing a knowledge base having information of said target stored therein, and said step of generating said sequence of test program instructions comprises selecting members of said sequence of test program instructions in accordance with said information in said knowledge base, wherein said step of selecting members is biased by said set of statements.

23. (Original) The computer software product according to claim 21, wherein at least a portion of said sequence of test program instructions are randomly generated.

24. (Canceled)

25. (Currently amended) The computer software product according to claim 21, wherein each of said ~~event~~ events ~~has~~ have an identifying name attribute.

26. (Currently amended) The computer software product according to claim 21, wherein each of said ~~event has~~ events have a triggering condition attribute.

27. (Currently amended) The computer software product according to claim 21, wherein each of said ~~event~~ events ~~comprises~~ comprise a body that is a template for generation of said alternate sequence of test program instructions.

28. (Currently amended) ~~The computer software product according to claim 21,~~ A computer software product, comprising a computer-readable medium in which computer program instructions are stored, which instructions, when read by a computer, cause the computer to generate test programs by performing the steps of:

accepting a set of statements, said set of statements including an event, wherein said event comprises an identifying name attribute, a triggering condition attribute, a priority value and a body;

responsive to said set of statements generating a sequence of test program instructions for a target;

while performing said step of generating said sequence of test program instructions determining if a condition of said event is satisfied; and

responsive to said step of determining generating an alternate sequence of test program instructions.

29. (Original) A computer software product, comprising a computer-readable medium in which computer program instructions are stored, which instructions, when read by a computer, cause the computer to generate test programs by performing the steps of:

defining a test program generation engine in a memory;

defining a design specification of a target in said memory, wherein said design specification comprises a knowledge base;

defining an architectural simulator of said target in said memory;

coupling said test program generation engine to said design specification;

coupling said test program generation engine to said architectural simulator;

accepting a set of statements into said test program generation engine, said set of statements including an event;

responsive to said set of statements, and to information in said knowledge base, causing said test program generation engine to generate a test program instruction that can be executed on said target;

thereafter determining in said architectural simulator whether a triggering condition of said event is satisfied by a simulated execution of said test program instruction;

in a first case, wherein said triggering condition is not satisfied, causing said test program generation engine to respond to said set of statements to generate a first sequence of test program instructions that can be executed on said target; and in a second case, wherein said triggering condition is satisfied, causing said test program generation engine to respond to an alternate set of statements of said event to generate a second sequence of test program instructions that can be executed on said target.

30. (Original) The computer software product according to claim 29, wherein at least a portion of said first sequence of test program instructions and said second sequence of test program instructions is generated randomly.

31. (Original) The computer software product according to claim 29, wherein said step of determining is performed by evaluating a state of said target prior to inclusion of said test program instruction in one of said first sequence of test program instructions and said second sequence of test program instructions.

32. (Original) The computer software product according to claim 31, wherein said step of evaluating said state is performed prior to said simulated execution of said test program instruction.

33. (Original) The computer software product according to claim 31, wherein said step of evaluating said state is performed subsequent to said simulated execution of said test program instruction.

34. (Original) The computer software product according to claim 31, wherein said step of evaluating said state is performed a first time prior to a simulated execution of said test program instruction and is performed a second time subsequent to said simulated execution thereof.

35. (Original) The computer software product according to claim 29, wherein said event comprises a plurality of events, each of said events having a priority value, and said step of determining is performed with respect to each of said events in an order of said priority value thereof.

36. (Original) The computer software product according to claim 29, wherein said set of statements is introduced into said test program generation engine as an input file.

37. (Original) The computer software product according to claim 29, wherein said event has an identifying name attribute.

38. (Original) The computer software product according to claim 29, wherein said event has a triggering condition attribute.

39. (Original) The computer software product according to claim 29, wherein said event comprises an input stream.

40. (Original) The computer software product according to claim 29, wherein said event comprises an identifying name attribute, a triggering condition attribute, a priority value and an input stream.

41. (Original) A test program generator, comprising:

a test program generation engine;

a design specification of a target, wherein said design specification comprises a knowledge base, wherein said test program generation engine is coupled to said design specification;

an architectural simulator of said target coupled to said test program generation engine;

wherein said test program generation engine is adapted to accept a set of statements, said set of statements including an event;

responsive to said set of statements, and to information in said knowledge base, said test program generation engine generates a test program instruction that can be executed on said target;

wherein said test program generation engine determines whether a triggering condition of said event is satisfied by a simulated execution of said test program instruction;

in a first case, wherein said triggering condition is not satisfied, responsive to said set of statements said test program generation engine generates a first sequence of test program instructions that can be executed on said target; and

in a second case, wherein said triggering condition is satisfied, responsive to an alternate input stream in said event said test program generation engine, generates a second sequence of test program instructions that can be executed on said target.

42. (Original) The test program generator according to claim 41, further comprising a design simulator for simulating said simulated execution.

43. (Original) The test program generator according to claim 41, wherein at least a portion of said first sequence of test program instructions and said second sequence of test program instructions is generated randomly.

44. (Original) The test program generator according to claim 41, wherein said architectural simulator determines whether said triggering condition of said event is satisfied by evaluating a state of said target prior to inclusion of said test program instruction in one of said first sequence of test program instructions and said second sequence of test program instructions.

45. (Original) The test program generator according to claim 44, wherein said step of evaluating said state is performed prior to said simulated execution of said test program instruction.

46. (Original) The test program generator according to claim 44, wherein said step of evaluating said state is performed subsequent to said simulated execution of said test program instruction.

47. (Original) The test program generator according to claim 44, wherein said step of evaluating said state is performed a first time prior to said simulated execution of said test program instruction and is performed a second time subsequent to said simulated execution.

48. (Original) The test program generator according to claim 41, wherein said event comprises a plurality of events, each of said events having a priority value, and said step of determining is performed with respect to each of said events in an order of said priority value thereof.

49. (Original) The test program generator according to claim 41, wherein said set of statements is introduced into said test program generation engine as an input file.

50. (Original) The test program generator according to claim 41, wherein said event has an identifying name attribute.

51. (Original) The test program generator according to claim 41, wherein said event has a triggering condition attribute.

52. (Original) The test program generator according to claim 41, wherein said event comprises an input stream entity.

53. (Original) The test program generator according to claim 41, wherein said event comprises an identifying name attribute, a triggering condition attribute, a priority value and an input stream entity.